University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: March 1st, 2023

Group Number: 198

Name	Student Number	CS Alias (Userid)	Preferred Email address
Danjiro Turner Okazaki	43562461	q7y2b	danjiro@student.ubc.ca
Olivia Pang	6660257	t1j3b	oliviaxpang@gmail.com
Bryan Lu	79152609	k4w4f	bryanzlu18@gmail.com

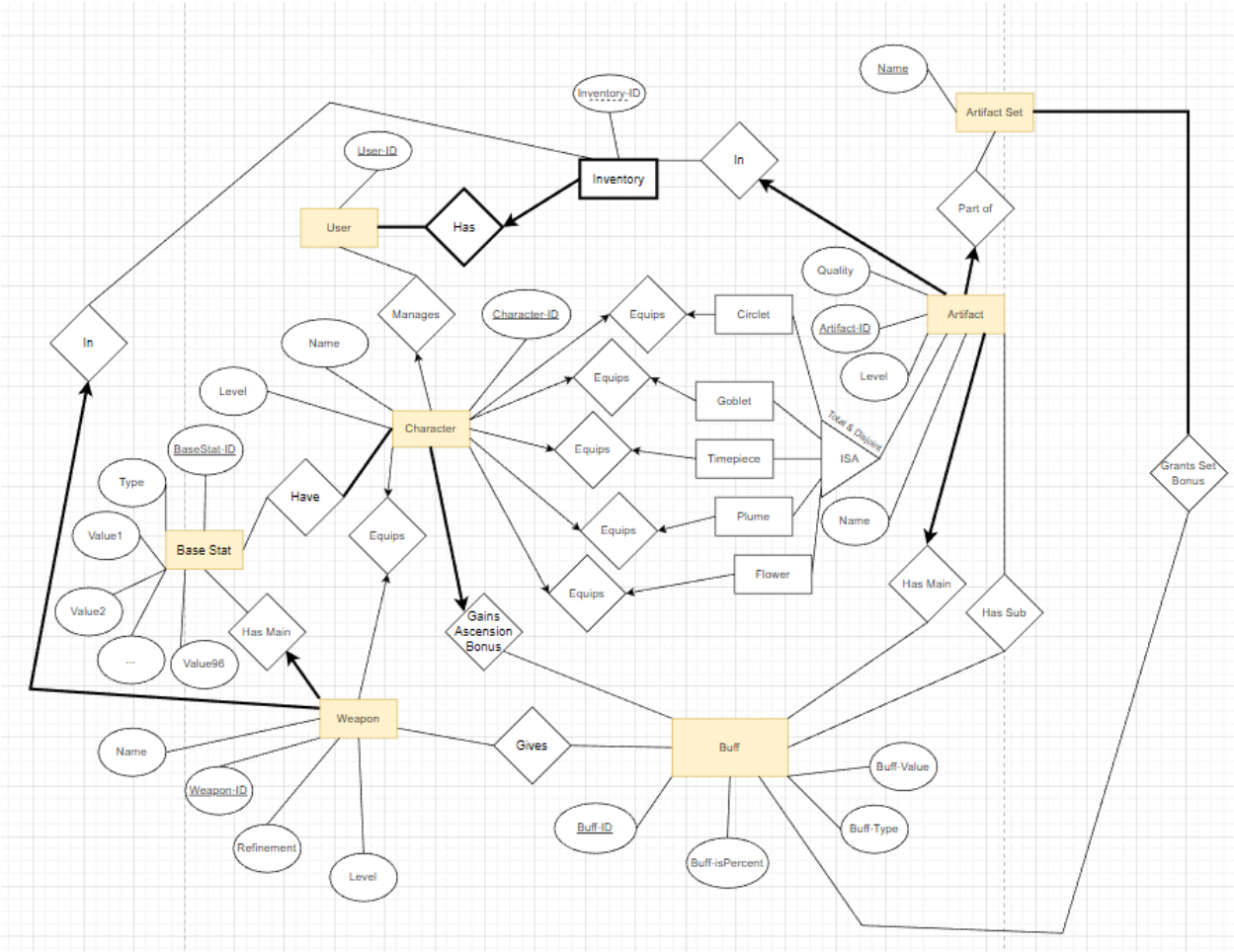
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Summary

GenshinDB is an application that helps players of the game Genshin Impact optimize their character builds by letting them see what a character's stats would be when equipped with items — weapons and artifacts — of the user's choice. In Genshin Impact, players can find, buy and craft items and equip one weapon and up to five artifacts to characters to increase their stats. There are five types of artifacts but characters can only carry one artifact of each type. As well, specific combinations of artifacts, called artifact sets, provide bonus stats. The game's UI is inefficient because players have to switch between multiple interfaces to see changes in stats. Additionally, in the game, if a player doesn't have an item they won't be able to see what their character's stats would be with that item equipped, whereas in this app players can model a character's stats with any items regardless if they actually have them. Thus, GenshinDB is a tool for efficiently visualizing stats while trying different combinations of items. The application will support: registering, adding and deleting characters and items, inventory search, equipping characters, and saving and loading the inventory.

ER Diagram Updates

- Added an attribute to weak entity **Inventory** to complete its key.
- Added total participation constraint on **Character** entity end of the **Have** relationship between **Base Stat** and **Character** because characters must have base stats.
- Renamed Stats to **Buff** and replaced its ISA relationships with a boolean attribute isPercent. This is a cleaner representation of the previous setup.
- Removed the Has Secondary relationship between **Weapon** and **Buff** and changed **Gives** relationship between **Weapon** and **Buff** to many-to-many. The Has Secondary relationship was redundant.
- Added ID attribute to **Base Stat** entity for use as primary key so foreign keys to **Base Stat** entities won't be two attributes and is easier to identify at a glance. We will have many many **Base Stat** foreign keys, so this will significantly reduce the amount of storage needed.
- Added ID attribute to **Character** so that different users could manage characters of the same name.
- Added ID attribute to **Buff** to simplify foreign keys.
- Changed the **Manages** relationship between **User** and **Character** to be many-to-one, one being **User**, so that each **Character** setup is only attributed to one **User**.
- Added many value attributes to **Base Stat** entity to drastically reduce the number of base stat entities needed and therefore also **Base Stat** foreign keys needed.
- Renamed "ID" attributes (eg. "Inventory-ID") of all entities for clarity and consistency.



Schema & FDs:

Relationships from the ER Diagram listed for convenience. Entities with total participation required are bolded.

- **Inventory** many to 1 **User** (Has)

- **Character** many to many Base Stat (Have)
- Character many to 1 User (Manages)
- **Character** many to 1 Buff (Gains Ascension Bonus)
- Character 1 to 1 Weapon (Equips)
- Character 1 to 1 of each Artifact subtype (Circlet, Goblet, Timepiece, Plume, Flower) (Equips)

- **Weapon** many to 1 Inventory (In)
- **Weapon** many to 1 Base Stat (Has Main)

- Buff many to many Weapon (Gives)
- Buff many to many **Artifact Set** (Grants Set Bonus)

- Artifact many to many Buff (Has Sub)
- **Artifact** many to 1 Buff (Has Main)

- **Artifact** many to 1 Inventory (In)
- **Artifact** many to 1 Artifact Set (Part of)

Because of the nature of our application, each entity must be identified uniquely only by its respective ID, therefore no Candidate Keys are listed.

3NF Decomposition is in green.

Inventory(Inventory-ID: INT, **User-ID**: INT)
Weak entity dependent on User entity.
FDs:
- No non-trivial FDs

Inventory(Inventory-ID: INT, **User-ID**: INT)

User(UID: INT)
FDs:
- No non-trivial FDs

User(UID: INT)

BaseStat(Value: INT, Type: VARCHAR(20))
FDs:
- No non-trivial FDs

BaseStat(BaseStat-ID: VARCHAR(50), Value1: INT, Value2: INT, …, Value96: INT, Type: VARCHAR(20))
Remade to normalize Weapon, see below
**96 attributes from Value1…Value96*
New FDs:
- BaseStat-ID → Value1, Value2, …, Value96, Type

Have(**BaseStat-Value**: INT, **BaseStat-Type**: VARCHAR(20), **Character-ID**: INT)
FDs:
- No non-trivial FDs

Have(**BaseStat-ID**: VARCHAR(50), **Character-ID**: INT)
Remade to normalize Weapon, see below
New FDs:
- No non-trivial FDs

Buff(Buff-ID: INT, Buff-Value: FLOAT, Buff-Type: VARCHAR(20), Buff-IsPercent: BIT)
Constraints: Buff-Value is NOT NULL, Buff-Type is NOT NULL, Buff-IsPercent is NOT NULL
**BIT is meant to be used as a Boolean*
FDs:
- Buff-ID → Buff-Value, Buff-Type, Buff-IsPercent

Buff(Buff-ID: INT, Buff-Value: FLOAT, Buff-Type: VARCHAR(20), Buff-IsPercent: BIT)

Gives(**Weapon-ID**: INT, **Buff-ID**: INT)

FDs:

- No non-trivial FDs

Gives(**Weapon-ID**: INT, **Buff-ID:** INT)

Weapon(Weapon-ID: INT, Name: VARCHAR(50), Refinement: INT, Level: INT, **BaseStat-Value**: INT, **BaseStat-Type**: VARCHAR(20), **Inventory-ID**: INT, **User-ID**: INT)
Constraints: Name is NOT NULL, BaseStat-Value is NOT NULL, BaseStat-Type is NOT NULL, Inventory-ID NOT NULL, and User-ID is NOT NULL
FDs:

- Weapon-ID → Name, Refinement, Level, BaseStat-ID, Inventory-ID, User-ID
- Name → BaseStat-Value, BaseStat-Type

R1:(Name, BaseStat-Value, BaseStat-Type), R':(Weapon-ID, Refinement, Level, Inventory-ID)

Weapon(Weapon-ID: INT, Name: VARCHAR(50), Refinement: INT, Level: INT, **BaseStat-ID:** VARCHAR(50), **Inventory-ID**: INT, **User-ID**: INT)
New Constraints: Name is NOT NULL, BaseStat-ID is NOT NULL, Inventory-ID NOT NULL, and User-ID is NOT NULL
FDs:

- Weapon-ID → Name, Refinement, Level, BaseStat-ID, Inventory-ID, User-ID

HasSub(**Artifact-ID**: INT, **Buff-ID**: INT)
Constraints: Buff-ID is NOT NULL.
FDs:

- No non-trivial FDs

HasSub(**Artifact-ID**: INT, **Buff-ID**: INT)

Artifact(Artifact-ID: INT, Quality: INT, Level: INT, Name: VARCHAR(40), **Buff-ID**: INT, **Inventory-ID**: INT, **User-ID**: INT, **ArtifactSet-Name**: VARCHAR(40))
Constraints: Name is NOT NULL, Buff-ID is NOT NULL, Inventory-ID is NOT NULL, User-ID is NOT NULL, ArtifactSet-Name is NOT NULL
FDs:

- Artifact-ID → Inventory-ID, Quality, Level, Name, Buff-ID, Inventory-ID, User-ID, ArtifactSet-Name
- Name → ArtifactSet-Name

R1:(Inventory-ID, User-ID), R':(Artifact-ID, Quality, Level, Name, Buff-Value, Buff-Type, Buff-IsPercent, Inventory-ID, ArtifactSet-Name)
R2:(Name, Buff-Type), R'':(Artifact-ID, Quality, Level, Name, Buff-Value, Buff-IsPercent, Inventory-ID, ArtifactSet-Name)
R3:(Name, Buff-IsPercent), R''':(Artifact-ID, Quality, Level, Name, Buff-Value, ArtifactSet-Name)
R4:(Name, ArtifactSet-Name), R'''':(Artifact-ID, Quality, Level, Name, Buff-Value)

R1(Inventory-ID, User-ID) - Same as Inventory
R2(Name, Buff-Type, Buff-IsPercent, ArtifactSet-Name)
R3(Artifact-ID, Quality, Level, Name, Buff-Value)

Circlet(**Artifact-ID**: INT)
ISA Artifact
FDs:

- No non-trivial FDs

Circlet(**Artifact-ID**: INT)

Goblet(**Artifact-ID**: INT)
ISA Artifact
FDs:

- No non-trivial FDs

Goblet(**Artifact-ID**: INT)

Timepiece(**Artifact-ID**: INT)
ISA Artifact
FDs:

- No non-trivial FDs

Timepiece(**Artifact-ID**: INT)

Plume(**Artifact-ID**: INT)
ISA Artifact
FDs:

- No non-trivial FDs

Plume(**Artifact-ID**: INT)

Flower(**Artifact-ID**: INT)
ISA Artifact
FDs:

- No non-trivial FDs

Flower(**Artifact-ID**: INT)

ArtifactSet(Name: VARCHAR(40))
FDs:

- No non-trivial Fds

ArtifactSet(Name: VARCHAR(40))

GrantsSetBonus(**ArtifactSet-Name**: VARCHAR(40), **Buff-ID**: INT)
Constraints: Buff-ID is NOT NULL
FDs:

- No non-trivial FDs

GrantsSetBonus(**ArtifactSet-Name**: VARCHAR(40), **Buff-ID**: INT)

Character(Character-ID: INT, Name: VARCHAR(30), Level: INT, **User-ID**: INT, **BaseStat-ID**: VARCHAR(50), **Buff-ID**: INT, **Weapon-ID**: INT, **Circlet-ID**: INT, **Goblet-ID**: INT, **Timepiece-ID**: INT, **Plume-ID**: INT, **Flower-ID**: INT)
Constraints: Name is NOT NULL, Buff-ID is NOT NULL, Weapon-ID is UNIQUE, Circlet-ID is UNIQUE, Goblet-ID is UNIQUE, Timepiece-ID is UNIQUE, Plume-ID is UNIQUE, Flower-ID is UNIQUE
FDs:

- Character-ID → Name, Level, User-ID, Buff-ID, Weapon-ID, Circlet-ID, Goblet-ID, Timepiece-ID, Plume-ID, Flower-ID

SQL DDL & Populating Tuples

```
CREATE TABLE Inventory(

      inventory-ID          INT,
      user-ID               INT,
      PRIMARY KEY (inventory-ID, user-ID),
      FOREIGN KEY (user-ID)
            REFERENCES user(user-ID)
             ON DELETE CASCADE
             ON UPDATE CASCADE)

INSERT
INTO        Inventory(inventory-ID, user-ID)
VALUES      (`1234567890`, `0987654321`),
            (`1234567891`, `1987654321`),
            (`1234567892`, `2987654321`),
            (`1234567893`, `3987654321`),
            (`1234567894`, `4987654321`)

CREATE TABLE User(

      user-ID          INT   PRIMARY KEY)

INSERT
INTO        User(user-ID)
VALUES      `0987654321`,
            `1987654321`,
            `2987654321`,
            `3987654321`,
            `4987654321`
```

```
CREATE TABLE BaseStat(     //unsure of syntax for this case

     baseStat-ID     VARCHAR(50)     PRIMARY KEY,
     value1          INT,
     value2          INT,
     …               INT,
     value96         INT,
     type            VARCHAR(20))

INSERT
INTO       BaseStat(baseStat-ID, value1 … value96, type)
VALUES     (`YelanATK`, `25`, …, `255`, `Attack`),
           (`Mistsplitter`, `45`, …, `565`, `Attack`),
           (`TheFlute`, `25`, …, `510`, `Energy Recharge`),
           (`XianglingDEF`, `35`, …, `540`, `Defense`),
           (`FischlHP`, `600`, …, `9000`, `HP`)

CREATE TABLE Have(

     baseStat-ID     INT  NOT NULL,
     character-ID    INT  NOT NULL,
     PRIMARY KEY (baseStat-ID, character-ID),
     FOREIGN KEY (baseStat-ID)
          REFERENCES BaseStat(baseStat-ID)
           ON DELETE NO ACTION
           ON UPDATE CASCADE
     FOREIGN KEY (character-ID)
          REFERENCES Character(character-ID)
           ON DELETE CASCADE
           ON UPDATE CASCADE)

INSERT
INTO       Have(baseStat-ID, character-ID)
VALUES     (`HutaoHP`, `1`),
           (`HutaoDEF`, `1`),
           (`HutaoATK`, `1`),
           (`XingqiuATK`, `4`),
           (`XingqiuDEF`, `4`)

CREATE TABLE Buff(

     buff-ID              INT             PRIMARY KEY,
     buff-value           FLOAT           NOT NULL,
     buff-type            VARCHAR(20)     NOT NULL,
     buff-isPercent       BIT             NOT NULL)
```

```
INSERT
INTO       Buff(buff-ID, buff-value, buff-type, buff-isPercent)
VALUES     (`1`, `124.0`, `Attack`, `0`),
           (`2`, `234.0`, `Defense`, `0`),
           (`3`, `23.5`, `Energy Recharge`, `1`),
           (`4`, `12.7`, `CRIT Rate`, `1`),
           (`5`, `130`, `HP`, `0`)

CREATE TABLE Gives(

      weapon-ID            INT,
      buff-ID              INT,
      PRIMARY KEY (weapon-ID, buff-ID),
      FOREIGN KEY (weapon-ID)
            REFERENCES Weapon(weapon-ID)
                  ON DELETE CASCADE
                  ON UPDATE CASCADE
      FOREIGN KEY (buff-ID)
            REFERENCES Buff(buff-ID)
                  ON DELETE CASCADE
                  ON UPDATE CASCADE)


INSERT
INTO       Gives(weapon-ID, buff-ID)
VALUES     (`1`, `12`),
           (`2`, `234`),
           (`3`, `233`),
           (`4`, `54`),
           (`5`, `90`)

CREATE TABLE Weapon(

      weapon-ID            INT             PRIMARY KEY,
      name                 VARCHAR(50)     NOT NULL,
      refinement           INT,
      level                INT,
      baseStat-ID          VARCHAR(50)     NOT NULL,
      inventory-ID         INT             NOT NULL,
      user-ID              INT             NOT NULL,
      FOREIGN KEY (baseStat-ID)
            REFERENCES BaseStat(baseStat-ID)
             ON DELETE NO ACTION
             ON UPDATE CASCADE
```

```
     FOREIGN KEY (inventory-ID, user-ID)
          REFERENCES Inventory(inventory-ID, user-ID),
           ON DELETE CASCADE
           ON UPDATE CASCADE)
```

INSERT

```
INTO       Weapon(weapon-ID, name, refinement, level, baseStat-ID,
inventory-ID, user-ID)
VALUES     (`1`, `Dragonspine Spear`, `1`, `1`, `DragonspineSpear`,
`1`, `1`),
           (`2`, `Deathmatch`, `2`, `2`, `Deathmatch`, `2`, `2`),
           (`3`, `Moon Piercer`, `3`, `3`, `MoonPiercer`, `3`, `3`),
           (`4`, `Black Tassle`, `4`, `4`, `Black Tassle`, `4`, `4`),
           (`5`, `Skyward Spine`, `5`, `5`, `Skyward Spine`, `5`,
`5`)
```

CREATE TABLE HasSub(

```
     artifact-ID          INT,
     buff-ID              INT              NOT NULL,
     PRIMARY KEY (artifact-ID, buff-ID),
     FOREIGN KEY (artifact-ID)
          REFERENCES Artifact(artifact-ID)
           ON DELETE CASCADE
     FOREIGN KEY (buff-ID)
          REFERENCES Buff(buff-ID),
           ON DELETE NO ACTION
           ON UPDATE CASCADE)
```

INSERT

```
INTO       HasSub(artifact-ID, buff-ID)
VALUES     (`12`, `2`),
           (`234`, `15`),
           (`233`, `1`),
           (`54`, `1`),
           (`644`, `200`)
```

CREATE TABLE Artifact(

```
     artifact-ID          INT                   PRIMARY KEY,
     quality              INT,
     level                INT,
     name                 VARCHAR(40)           NOT NULL,
     buff-ID              INT                   NOT NULL,
     inventory-ID         INT                   NOT NULL,
```

```
     user-ID              INT                    NOT NULL,
     artifactSet-name          VARCHAR(40)          NOT NULL,
     FOREIGN KEY (buff-id)
          REFERENCES Buff(buff-ID),
           ON DELETE NO ACTION
           ON UPDATE CASCADE
     FOREIGN KEY (inventory-ID)
          REFERENCES Inventory(inventory-ID, user-ID)
           ON DELETE CASCADE
           ON UPDATE CASCADE
     FOREIGN KEY (artifactSet-name)
          REFERENCES ArtifactSet(name)
           ON DELETE CASCADE
           ON UPDATE CASCADE)
```

```
INSERT
INTO       Artifact(artifact-ID, quality, level, name,
                    buff-id)
VALUES     (`1`, `1`, `1`, `Flower of Life`, `12`),
           (`2`, `2`, `2`, `Plume of Death`, `25`),
           (`3`, `3`, `3`, `Sands of Eon`, `311`),
           (`4`, `4`, `4`, `Goblet of Eonothem`, `44`),
           (`5`, `5`, `5`, `Circlet of Logos`, `500`)
```

```
CREATE TABLE ArtifactSet(

     name       VARCHAR(40)     PRIMARY KEY)
```

```
INSERT
INTO       ArtifactSet(name)
VALUES     `Wanderer's Troupe`,
           `Lucky Dog`,
           `The Exile`,
           `Maiden Beloved`,
           `Instructor`
```

```
CREATE TABLE Circlet(

     artifact-ID     INT        PRIMARY KEY,
     FOREIGN KEY (artifact-ID)
          REFERENCES Artifact(artifact-ID)
           ON DELETE CASCADE
           ON UPDATE CASCADE)
```

```
INSERT
```

```
INTO       Circlet(artifact-ID)
VALUES     `1`,
           `2`,
           `3`,
           `4`,
           `5`
```

```
CREATE TABLE Goblet(

     artifact-ID     INT        PRIMARY KEY,
     FOREIGN KEY (artifact-ID)
          REFERENCES Artifact(artifact-ID)
           ON DELETE CASCADE
           ON UPDATE CASCADE)
```

```
INSERT
INTO       Goblet(artifact-ID)
VALUES     `134`,
           `21`,
           `34`,
           `45`,
           `567`
```

```
CREATE TABLE Timepiece(

     artifact-ID     INT        PRIMARY KEY,
     FOREIGN KEY (artifact-ID)
          REFERENCES Artifact(artifact-ID)
           ON DELETE CASCADE
           ON UPDATE CASCADE)
```

```
INSERT
INTO       Timepiece(artifact-ID)
VALUES     `156`,
           `262`,
           `326`,
           `4234`,
           `52`
```

```
CREATE TABLE Plume(

     artifact-ID     INT        PRIMARY KEY,
     FOREIGN KEY (artifact-ID)
          REFERENCES Artifact(artifact-ID)
           ON DELETE CASCADE
```

```
            ON UPDATE CASCADE)

INSERT
INTO       Plume(artifact-ID)
VALUES     `234`,
           `6342`,
           `1235`,
           `79`,
           `56`

CREATE TABLE Flower(

     artifact-ID     INT        PRIMARY KEY,
     FOREIGN KEY (artifact-ID)
          REFERENCES Artifact(artifact-ID)
           ON DELETE CASCADE
           ON UPDATE CASCADE)

INSERT
INTO       Flower(artifact-ID)
VALUES     `234`,
           `459`,
           `3454`,
           `1345`,
           `2435`

CREATE TABLE GrantsSetBonus(

     artifactSet-name           VARCHAR(40),
     buff-ID              INT             NOT NULL,
     PRIMARY KEY (artifactSet-name, buff-ID),
     FOREIGN KEY (artifactSet-name)
          REFERENCES ArtifactSet(name)
           ON DELETE CASCADE
           ON UPDATE CASCADE
     FOREIGN KEY (buff-ID)
          REFERENCES Buff(buff-ID),
           ON DELETE NO ACTION
           ON UPDATE CASCADE)


INSERT
INTO       GrantsSetBonus(artifactSet-name, buff-ID)
VALUES     (`The Exile`, `124`),
           (`Lucky Dog`, `234`),
```

```
        (`Berserker`, `23`),
        (`Scholar`, `12`),
        (`Gladiator's Finale`, `130`)

CREATE TABLE Character(

    character-ID        INT         PRIMARY KEY,
    name                VARCHAR(30) NOT NULL,
    level               INT,
    user-ID             INT,
    Buff-ID             INT         NOT NULL,
    weapon-ID           INT         UNIQUE,
    circlet-ID          INT         UNIQUE,
    goblet-ID           INT         UNIQUE,
    timepiece-ID        INT         UNIQUE,
    plume-ID            INT         UNIQUE,
    flower-ID           INT         UNIQUE,
    FOREIGN KEY (user-ID)
        REFERENCES User(user-ID)
         ON DELETE CASCADE
    FOREIGN KEY (buff-ID)
        REFERENCES Buff(buff-ID),
         ON DELETE NO ACTION
         ON UPDATE CASCADE
    FOREIGN KEY (weapon-ID)
        REFERENCES Weapon(weapon-ID)
         ON DELETE SET NULL
    FOREIGN KEY (circlet-ID)
        REFERENCES Circlet(artifact-ID)
         ON DELETE SET NULL
    FOREIGN KEY (goblet-ID)
        REFERENCES Goblet(artifact-ID)
         ON DELETE SET NULL
    FOREIGN KEY (timepiece-ID)
        REFERENCES Timepiece(artifact-ID)
         ON DELETE SET NULL
    FOREIGN KEY (plume-ID)
        REFERENCES Plume(artifact-ID)
         ON DELETE SET NULL
    FOREIGN KEY (flower-ID)
        REFERENCES Flower(artifact-ID)
         ON DELETE SET NULL)

INSERT
INTO    Character(character-ID, name, level, user-ID,
```

```
                    buff-ID, weapon-ID, circlet-ID, goblet-ID,
                    timepiece-ID, plume-ID, flower-ID)
VALUES      (`1`, `Traveler`, `20`, `23746279`,
            `1`, `12`, `23`, `234`, `53`, `65`, `254`),
            (`2`, `Amber`, `20`, `23746279`,
            `2`, `13`, `24`, `235`, `54`, `66`, `255`),
            (`3`, `Noelle`, `20`, `23746279`,
            `3`, `14`, `25`, `236`, `55`, `67`, `256`),
            (`4`, `Kaeya`, `20`, `23746279`,
            `4`, `15`, `26`, `237`, `56`, `68`, `257`),
            (`5`, `Lisa`, `20`, `23746279`,
            `5`, `16`, `27`, `238`, `57`, `69`, `258`)
```